PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-325248

(43)Date of publication of application : 22.11.2001

(51)Int.Cl. G06F 17/21

(21)Application number : 2000-144947

(22)Date of filing : 17.05.2000

(71)Applicant : FUJI XEROX CO LTD

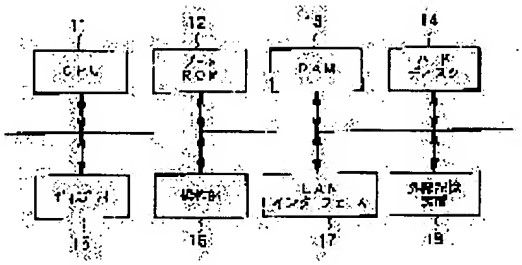
(72)Inventor : IWATA NOBUO

(54) DOCUMENT DATA PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a document data processor having improved processing efficiency in contrast to a conventional document data processor which has a problem that the processing efficiency is low since the processing of an HTML parser is performed after the processing of an XML parser is performed.

SOLUTION: In this document data processor, a CPU 11 reads document data and performs the processing as the XML parser. At the time of detecting a tag which is not an XML tag during the processing of the XML parser, the start tag processing part or end tag processing part of the HTML parser is activated and the pertinent part is processed. Further, at the time of finding the tag related to CDATA or a pre-format, a corresponding processing is performed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of

rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2001－325248
（Ｐ2001－325248Ａ）
(43)公開日　平成13年11月22日(2001.11.22)

(51)Int.Cl.'	識別記号	Ｆ Ｉ	ﾃｰﾏｺｰﾄﾞ(参考)
Ｇ０６Ｆ　17/21	５０１	Ｇ０６Ｆ　17/21	５０１Ｔ　５Ｂ００９

審査請求　未請求　請求項の数９　ＯＬ　（全 10 頁）

(21)出願番号　特願2000－144947(Ｐ2000－144947)

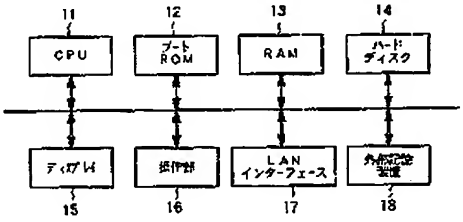
(22)出願日　平成12年５月17日(2000.5.17)

(71)出願人　000005496
富士ゼロックス株式会社
東京都港区赤坂二丁目17番22号
(72)発明者　岩田　伸夫
神奈川県海老名市本郷2274番地　富士ゼロックス株式会社海老名事業所内
(74)代理人　100075258
弁理士　吉田　研二　（外２名）

Ｆターム(参考)　５Ｂ００９　ＮＡ０５

(54)【発明の名称】　文書データ処理装置

(57)【要約】
【課題】　従来の文書データ処理装置では、ＸＭＬパーザの処理が行われた後にＨＴＭＬパーザの処理が行われるので、処理効率が低いという問題点があったが、本発明では、処理効率を向上できる文書データ処理装置を提供する。
【解決手段】　ＣＰＵ１１が文書データを読み込んで、ＸＭＬパーザとしての処理を行い、このＸＭＬパーザの処理中に、ＸＭＬタグでないタグを検出すると、ＨＴＭＬパーザの開始タグ処理部又は終了タグ処理部を起動して当該部分を処理し、さらにＣＤＡＴＡ又はプレフォーマットに関連するタグであるときには、対応する処理を行う文書データ処理装置である。



1

【特許請求の範囲】

【請求項１】　ＨＴＭＬにより記述された部分文書データとＸＭＬにより記述された部分文書データの少なくとも一方を含むタグ付き文書データ、

を処理する文書データ処理装置であって、

所定の部分文書データをＨＴＭＬ文書として処理し、出力するＨＴＭＬパーザ手段と、

処理対象となったタグ付き文書データを部分文書データごとに順次解析し、解析の結果に応じて前記ＨＴＭＬパーザ手段を起動する動作と、当該解析された部分文書データをＸＭＬ文書として処理する動作とのいずれかを選択的に行うＸＭＬパーザ手段と、

を含むことを特徴とする文書データ処理装置。

【請求項２】　請求項１に記載の文書データ処理装置であって、さらに、

処理対象となったタグ付き文書データのデータ型属性を取得する手段と、

前記データ型属性に対応付けて事前に定義されているデフォルトデータ型定義を取得する手段とを含み、

前記ＸＭＬパーザ手段が、前記取得したデフォルトデータ型定義に従って処理を行うことを特徴とする文書データ処理装置。

【請求項３】　請求項２に記載の文書データ処理装置であって、

前記ＸＭＬパーザ手段は、処理対象となったタグ付き文書データにＸＭＬ宣言がない場合にのみ前記取得したデフォルトデータ型定義に従って処理を行うことを特徴とする文書データ処理装置。

【請求項４】　請求項１に記載の文書データ処理装置であって、

処理対象となったタグ付き文書データと、当該文書データに関連づけられたデータ型定義とを取得する手段と、

前記タグ付き文書データのデータ型属性を取得する手段と、

前記データ型属性に対応付けて事前に定義されているデフォルトデータ型定義を取得する手段とを含み、

前記ＸＭＬパーザ手段が、前記取得したデータ型定義とデフォルトデータ型定義とに従って処理を行うことを特徴とする文書データ処理装置。

【請求項５】　請求項１から４のいずれかに記載の文書データ処理装置において、

特殊タグごとに、当該特殊タグに関連する部分文書データを処理するパーザ手段と、

前記特殊タグの情報と、当該特殊タグに対応するパーザ手段とを少なくとも一組設定する手段を含み、

前記ＸＭＬパーザ手段は、タグ付き文書データを解析し、当該解析の結果、前記特殊タグを検出すると、当該特殊タグに対応するパーザ手段を起動することを特徴とする文書データ処理装置。

【請求項６】　請求項５に記載の文書データ処理装置において、

前記特殊タグには、少なくともＣＤＡＴＡに関連するタグと、プレフォーマットに関連するタグとのいずれかを含むことを特徴とする文書データ処理装置。

【請求項７】　請求項１から６のいずれかに記載の文書データ処理装置において、

さらに、各タグごとの省略可否情報を格納する手段を含み、

前記ＸＭＬパーザ手段は、省略可能に設定されたタグを検出すると、タグが省略されているか否かを解析し、当該解析の結果に基づいてタグが省略されているときには、当該省略されたタグを補完して文書データを処理することを特徴とする文書データ処理装置。

【請求項８】　第１のルールで記述された部分文書データと第２のルールに従って記述された部分文書データの少なくとも一方を含む文書データ、

を処理する文書データ処理装置であって、

所定の部分文書データを前記第１のルールに従って処理し、出力する第１パーザ手段と、

処理対象となった文書データを部分文書データごとに順次解析し、解析の結果に応じて前記第１パーザ手段を起動する動作と、当該解析された部分文書データを第２のルールに従って処理する動作とのいずれかを選択的に行う第２パーザ手段と、

を含むことを特徴とする文書データ処理装置。

【請求項９】　第１のルールで記述された部分文書データと第２のルールに従って記述された部分文書データの少なくとも一方を含む文書データ、

を処理する文書データ処理プログラムであって、

所定の部分文書データを前記第１のルールに従って処理し、出力する第１パーザモジュールと、

処理対象となった文書データを部分文書データごとに順次解析し、解析の結果に応じて前記第１パーザモジュールを起動する動作と、当該解析された部分文書データを第２のルールに従って処理する動作とのいずれかを選択的に行う第２パーザモジュールと、

を含む文書データ処理プログラムを格納したことを特徴とするコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、構造化された文書データを表示等するための文書データ処理装置に係り、特にＸＭＬ（Extensible Markup Language）と、ＨＴＭＬ（HyperText Markup Language）との双方に対応可能な文書データ処理装置に関する。

【０００２】

【従来の技術】近年、インターネットで広く用いられる文書データの形式としてＨＴＭＬと呼ばれるマーク付け言語がある。ＨＴＭＬは、ＳＧＭＬ（Standard Generalized Markup Language）と呼ばれる、メタ言語文法に準

錠して記述されている。しかし、このＳＧＭＬにより記述された文書の処理は、種々の指定処理が可能であるものの、処理が複雑になるため、ＸＭＬと呼ばれる簡略化された言語が策定されつつある。また、このＸＭＬには、文書中に文書の属性や特定の処理の対象となるデータ（直接表示はされないが処理装置で利用されるデータ）を含ませることができる。

【０００３】このＸＭＬにより記述された文書データは、ＨＴＭＬと同様に取得されて処理されるのであるが、従来の文書データ処理装置としてのＷｅｂブラウザを含むパーソナルコンピュータでは、このＸＭＬを直接処理できないため、取得したＸＭＬ文書を、ＨＴＭＬに変換する処理を行ってから、ＨＴＭＬ文書として改めて処理を行っている。

【０００４】ところで、ＸＭＬやＳＧＭＬでは、文書ファイル中または文書ファイルから参照される他のファイルにＤＴＤ（Data Type Difinition）を設定してシンタックスを宣言し、この宣言に従って文書データの処理を行わせる。

【０００５】例えば、ＳＧＭＬでは、マーク付けのためのタグ（特定の処理を行わせるべきコンテンツを囲むデータ）をＤＴＤを用いて定義できる。このＳＧＭＬでは、終了を表すタグを省略して記載できるようにＤＴＤを用いた定義を作成できる。このように終了のタグを省略した文書データを処理する場合、当該タグに対するコンテンツであるか否かを判別し、そうでないコンテンツが見いだされた場合に、終了タグがあったものとして処理する必要がある。具体的にＨＴＭＬの<P>（段落の開始タグ）は、対応する終了タグ（</P>）を省略できることをＤＴＤを用いて定義できる。さらにＳＧＭＬでは、文書データ中に必須の開始タグを省略可能と定義することもできるようになっている。

【０００６】これに比べ、ＸＭＬは、開始タグと終了タグの種類が制限されるとともに、コンテンツのタイプも参照データとタグとの関係でＳＧＭＬより限定的に定められている。すなわち、ＳＧＭＬでは、コンテンツ中のタグや（画像データなどに対する）参照データを処理するＰＣＤＡＴＡ（Parsed Character Data）と、タグを無視して参照データのみを処理するＲＣＤＡＴＡ（Replaceable Character Data）と、タグも参照データも単なる文字列として扱うことができるＣＤＡＴＡ（Character Data）との３種類があったが、ＸＭＬでは、ＰＣＤＡＴＡのみが利用可能である。ここで、ＣＤＡＴＡは、表示や印刷の目的でなく、ＪａｖａＳｃｒｉｐｔによるスクリプトプログラムの記述に利用されている。

【０００７】このため、従来の文書データ処理装置では、ＸＭＬとＨＴＭＬの双方に対応させるために、ＸＭＬデータを処理してＨＴＭＬデータに変換し、次に変換後のＨＴＭＬデータを処理することとしている。

【０００８】

【発明が解決しようとする課題】このように、上記従来の文書データ処理装置では、ＸＭＬをＨＴＭＬへ変換するためにＸＭＬパーザを実行し、さらにこのパーザの処理結果に基づいてＨＴＭＬパーザを実行するので、処理負荷が大きくなる。また、既存のＨＴＭＬ文書の多くは、ＤＴＤ定義が関連づけられていないものが多く、ＸＭＬパーザが正常に動作しない場合がある。さらに、既存のＨＴＭＬ文書では、終了タグが省略されているものが多く、ＸＭＬパーザがそのまま正常に処理できないという問題点があった。

【０００９】さらに、近年のＨＴＭＬ文書データには、ＪａｖａＳｃｒｉｐｔのための<SCRIPT>タグや、スタイルシートの指定のための<STYLE>タグ等が設定されており、コンテンツがＣＤＡＴＡとして処理されることが前提となっているものがある。さらに<PRE>タグと呼ばれるタグを用いて、スペースや改行をコンテンツ通りに表示する手法が用いられたものもある。これらの文書データは、ＸＭＬパーザではそのまま正常に処理することができない。

【００１０】本発明は上記実情に鑑みて為されたもので、ＨＴＭＬ中に混在するＸＭＬデータを正常に処理でき、かつ処理効率を向上できる文書データ処理装置を提供することを目的とする。

【００１１】

【課題を解決するための手段】上記従来例の問題点を解決するための請求項１記載の発明は、文書データ処理装置であって、ＨＴＭＬにより記述された部分文書データとＸＭＬにより記述された部分文書データの少なくとも一方を含むタグ付き文書データ、を処理する文書データ処理装置であって、所定の部分文書データをＨＴＭＬ文書として処理し、出力するＨＴＭＬパーザ手段と、処理対象となったタグ付き文書データを部分文書データごとに順次解析し、解析の結果に応じて前記ＨＴＭＬパーザ手段を起動する動作と、当該解析された部分文書データをＸＭＬ文書として処理する動作とのいずれかを選択的に行うＸＭＬパーザ手段と、を含むことを特徴としている。

【００１２】このＸＭＬパーザ手段により、ＸＭＬパーザの解析処理中に検出されたＨＴＭＬ部分の文書データがＨＴＭＬパーザにより処理され、処理効率を向上できる。また、上記従来例の問題点を解決するための請求項２記載の発明は、請求項１記載の文書データ処理装置において、処理対象となったタグ付き文書データのデータ型属性を取得する手段と、前記データ型属性に対応付けて事前に定義されているデフォルトデータ型定義を取得する手段とを含み、前記ＸＭＬパーザ手段が、前記取得したデフォルトデータ型定義に従って処理を行うことを特徴としている。これにより、ＤＴＤ定義が関連づけられていない文書データに対してもデフォルトのＤＴＤを適用でき、正常に文書データを処理できる。

5

【００１３】さらに、ＸＭＬパーザ手段は、処理対象となったタグ付き文書データにＸＭＬ宣言がない場合にのみ前記取得したデフォルトデータ型定義に従って処理を行うことも好適である。

【００１４】また、上記従来例の問題点を解決するための請求項４記載の発明は、請求項１に記載の文書データ処理装置であって、処理対象となったタグ付き文書データと、当該文書データに関連づけられたデータ型定義とを取得する手段と、前記タグ付き文書データのデータ型属性を取得する手段と、前記データ型属性に対応付けて事前に定義されているデフォルトデータ型定義を取得する手段とを含み、前記ＸＭＬパーザ手段が、前記取得したデータ型定義とデフォルトデータ型定義とに従って処理を行うことを特徴としている。

【００１５】さらに、上記従来例の問題点を解決するための請求項５記載の発明は、請求項１から４のいずれかに記載の文書データ処理装置において、特殊タグごとに、当該特殊タグに関連する部分文書データを処理するパーザ手段と、前記特殊タグの情報と、当該特殊タグに対応するパーザ手段とを少なくとも一組設定する手段を含み、前記ＸＭＬパーザ手段は、タグ付き文書データを解析し、当該解析の結果、前記特殊タグを検出すると、当該特殊タグに対応するパーザ手段を起動することを特徴としている。

【００１６】また、ここで前記特殊タグには、少なくともＣＤＡＴＡに関連するタグと、プレフォーマットに関連するタグとのいずれかを含むことが好ましい。

【００１７】さらに、上記従来例の問題点を解決するための請求項７記載の発明は、請求項１から６のいずれかに記載の文書データ処理装置において、さらに、各タグごとの省略可否情報を格納する手段を含み、前記ＸＭＬパーザ手段は、省略可能に設定されたタグを検出すると、タグが省略されているか否かを解析し、当該解析の結果に基づいてタグが省略されているときには、当該省略されたタグを補完して文書データを処理することを特徴としている。これによりタグが省略された文書データに対しても正常な処理を行うことができる。

【００１８】上記従来例の問題点を解決するための請求項８記載の発明は、文書データ処理装置であって、第１のルールで記述された部分文書データと第２のルールに従って記述された部分文書データの少なくとも一方を含む文書データ、を処理する文書データ処理装置であって、所定の部分文書データを前記第１のルールに従って処理し、出力する第１パーザ手段と、処理対象となった文書データを部分文書データごとに順次解析し、解析の結果に応じて前記第１パーザ手段を起動する動作と、当該解析された部分文書データを第２のルールに従って処理する動作とのいずれかを選択的に行う第２パーザ手段と、を含むことを特徴としている。

【００１９】

6

【発明の実施の形態】本発明の実施の形態について図面を参照しながら説明する。本発明の実施の形態に係る文書データ処理装置は、パーソナルコンピュータであり、具体的には図１に示すように、ＣＰＵ１１と、ブートＲＯＭ１２と、ＲＡＭ１３と、ハードディスク１４と、ディスプレイ１５と、操作部１６と、ＬＡＮインタフェース１７と、外部記憶装置１８とから基本的に構成されている。

【００２０】ＣＰＵ１１は、電源投入直後にブートＲＯＭ１２に格納されているプログラムをＲＡＭ１３上にロードして実行する初期化処理を行う。この初期化処理により、ＣＰＵ１１は、ハードディスク１４に格納されたオペレーティングシステムをＲＡＭ１３上にロードし、処理を開始する。そして、ＣＰＵ１１は、操作部１６から入力される指示により、文書データの処理を行うブラウザをハードディスク１４からＲＡＭ１３上にロードして処理を行う。このブラウザの処理については、後に詳しく説明する。

【００２１】ブートＲＯＭ１２は、ＣＰＵ１１の初期化処理に関連するプログラムを格納している。ＲＡＭ１３は、ＣＰＵ１１のワークメモリとして動作する。ハードディスク１４は、ＣＰＵ１１が処理する各種プログラムを格納している。またこのハードディスク１４は、ＣＰＵ１１の処理に必要なデータ（例えば事前に設定されたデフォルトＤＴＤ等）を格納している。

【００２２】ディスプレイ１５は、ＣＰＵ１１から入力される指示により、種々のデータを表示出力する。操作部１６は、キーボードやマウス等であり、ユーザが行う操作の内容をＣＰＵ１１に伝達する。ＬＡＮインタフェース１７は、ＬＡＮ（LocalArea Network）又はインターネットを経由してＷｅｂサーバに接続されており、ＣＰＵ１１から入力される指示によりネットワークを介してデータを送信し、また、ネットワークを介して到来するデータを受信してＣＰＵ１１に出力する。

【００２３】外部記憶装置１８は、フロッピー（登録商標）ディスクや光磁気ディスク等、光学的又は電磁気的にデータを保持し、コンピュータにより読み取り可能な記録媒体等からデータを読み出してＣＰＵ１１に出力する。ＣＰＵ１１は、この外部記憶装置１８から読み出したデータをハードディスク１４に処理プログラムとしてインストールする。

【００２４】ここでＣＰＵ１１の文書データ処理について説明する。本実施の形態に係るＣＰＵ１１が処理する文書データ処理のためのブラウザプログラムは、図２に示すように、ＴＣＰ／ＩＰプロトコル解析部２１と、ＸＭＬパーズ部２２と、ＨＴＭＬパーズ部２３と、ブラウザコア部２４と、描画部２５とから構成されている。また、ＨＴＭＬパーズ部２３は、プレフォーマット処理部３１と、ＣＤＡＴＡ処理部３２と、開始タグ処理部３３と、終了タグ処理部３４とから構成されている。ここ

で．ＸＭＬパーズ部２２が．本発明のＸＭＬパーザ手段又は第２パーザ手段に、ＨＴＭＬパーズ部２３が本発明のＨＴＭＬパーザ手段又は第１パーザ手段にそれぞれ相当している。また、プレフォーマット処理部３１とＣＤＡＴＡ処理部３２とが本発明の特殊タグに対応するパーザ手段に相当し．プレフォーマット処理すべき文書データは．特殊タグ「<PRE>」に関連づけられ．ＣＤＡＴＡに関連する特殊タグは、「<SCRIPT>」等である。

【００２５】ＴＣＰ／ＩＰプロトコル処理部２１は、ＴＣＰ／ＩＰプロトコルによってネットワークを経由して文書データを取得し、ＸＭＬパーズ部２２に出力する。ここで、ＴＣＰ／ＩＰプロトコル処理部２１が取得する文書データは．具体的に図３で示すようなものである。この図３において、文字「<」と．「>」とで囲まれている部分（例えば先頭の<HTML>）がタグと呼ばれる。また．このタグのうち、「</」で始まるものが終了タグであり．そうでないものが開始タグである。図３に示すように．本実施の形態の文書データ処理装置において想定している文書データは、開始タグとコンテンツデータと終了タグとからなる基本構造が入れ子になっているものである。すなわち、開始タグ<HTML>と．終了タグ</HTML>の間のコンテンツには、さらに開始タグ<BODY>と、終了タグ</BODY>とに囲まれたコンテンツがあり、さらに．このコンテンツ内にも基本構造が複数含まれている。尚、この図３の文書データにおいては、数多く存在するＨＴＭＬ文書と同様に．例えば「
」タグに対応する終了タグが省略されている。

【００２６】また、図３には現れていないが、ＸＭＬにおいては、「/>」で終了するタグは．便宜的に終了タグとして扱われるのが一般的である。

【００２７】ここで、ＣＰＵ１１がＴＣＰ／ＩＰプロトコル処理部２１で取得した文書データに対して行うＸＭＬパーズ部２２としての処理を図４を参照して説明する。尚、以下の説明において、ハードディスク１４には．動作パラメータと、デフォルトデータ型定義に相当するデフォルトＤＴＤとが事前に設定され、格納されているものとする。ここで．動作パラメータとは．図５に示すように、開始タグ処理部３３へのポインタ（Ａ）と．終了タグ処理部３４へのポインタ（Ｂ）と．ＣＤＡＴＡ処理部３２へのポインタ（Ｃ）と．ＣＤＡＴＡとして処理すべきタグ名の配列（Ｄ）と．プレフォーマット処理部３１へのポインタ（Ｅ）と、プレフォーマット処理部３１で処理すべきタグ名の配列（Ｆ）と、終了タグの省略の可否を表すフラグ（Ｇ）とを関連づけたものであり．デフォルトＤＴＤは．要素宣言と、属性宣言とを含み．要素宣言は、図６（ａ）に示すように、識別子（Ｈ）と、タグ名（Ｉ）と．タイプ（Ｊ）と、開始タグ及び終了タグの省略可否を表すフラグ（Ｋ）と．コンテンツに含まれる可能性のあるタグのリスト（Ｌ）とを関連づけたものである。また．属性宣言は、図６（ｂ）に示すように、識別子（Ｍ）と、タグ名（Ｎ）と．属性名（Ｏ）と、属性値のタイプ（Ｐ）とを関連づけたものであり．属性値タイプが列挙型（enumeration）である場合には、さらに取りうる値の配列（Ｑ）が関連づけられている。これらの図５及び図６において配列やリストは．通常広く知られるように、ＮＵＬＬで配列の終了を識別することとしている。

【００２８】ＣＰＵ１１は．ＸＭＬパーズ部２２の処理として、図４に示すように．まず、動作パラメータをハードディスク１４からロードし（Ｓ１）、デフォルトＤＴＤをハードディスク１４からロードする（Ｓ２）。そして．文書データを読込み（Ｓ３）．文書データが終了したか否かを調べ（Ｓ４）．終了していれば（Ｙｅｓならば）、処理を終了する。

【００２９】また、処理Ｓ４において．終了していなければ（Ｎｏならば）、読み込んだ文書データがＸＭＬタグであるか否かを調べる（Ｓ５）。ここでＸＭＬタグとは．「<!」や．「<?」で開始する特別なタグである。このようなタグとして例えば．「<!—」で始まるコメントなどがある。この処理Ｓ５において．ＸＭＬタグであれば（Ｙｅｓならば）、ＸＭＬタグの処理を実行して（Ｓ６）．処理Ｓ３に戻って処理を続ける（Ａ）。一方、ＸＭＬタグでなければ（Ｎｏならば）．さらに開始タグであるか否かを調べ（Ｓ７）．開始タグであれば（Ｙｅｓならば）、当該開始タグのタグ名をキーとしてデフォルトＤＴＤを参照し、開始タグ処理部３３へのポインタを取得し、読み込んだ文書データを引数として開始タグ処理部３３を起動する（Ｓ８）。この開始タグ処理部３３の動作については後述する。

【００３０】そして、開始タグ処理部３３の処理が完了すると、ＣＰＵ１１は、ＸＭＬパーザの処理を再開し、処理Ｓ８で処理した開始タグがＣＤＡＴＡとして処理すべきタグであるか否かを動作パラメータを参照して検査し（Ｓ９）、ＣＤＡＴＡとして処理すべきタグであれば（Ｙｅｓであれば）、ＣＤＡＴＡ処理部３２を起動する（Ｓ１０）。ＣＰＵ１１は．このＣＤＡＴＡ処理部３２の動作として．対応する終了タグが読み込まれるまでの間．ＣＤＡＴＡとして処理し、木構造に追加する処理を行う。そして．ＣＤＡＴＡ処理部３２の動作が完了すると．ＣＰＵ１１は、処理Ｓ３に戻ってＸＭＬパーザの処理を続ける。

【００３１】一方、処理Ｓ９において．ＣＤＡＴＡとして処理すべきタグでなければ（Ｎｏであれば）．さらに当該開始タグがプレフォーマット処理すべきタグであるか否かを動作パラメータを参照して検査し（Ｓ１１）、プレフォーマット処理すべきタグであれば（Ｙｅｓならば）．プレフォーマット処理部３１を起動する（Ｓ１２）。そしてＣＰＵ１１は．プレフォーマット処理部３１の処理として．当該タグに対応する終了タグが読み込まれる間、ＰＣＤＡＴＡとして木構造に追加する処理を

行う。このプレフォーマット処理部３１では、改行コードを検出すると、改行タグ
に置き換える処理を行う。そして、ＣＰＵ１１は、プレフォーマット処理部３１の処理が完了すると、処理Ｓ３に戻ってＸＭＬパーザの処理を続ける。また、処理Ｓ１１において、プレフォーマット処理すべきタグでなければ、そのまま処理Ｓ３に戻ってＸＭＬパーザの処理を続ける。

【００３２】さらに、ＣＰＵ１１は、処理Ｓ７において、開始タグでなければ（Ｎｏならば）、読み込んだ文書データが終了タグであるか否かを調べる（Ｓ１３）。そして、終了タグであれば（Ｙｅｓならば）、終了タグ処理部３４を起動し（Ｓ１４）、終了タグ処理を行い、処理Ｓ３に戻って処理を続ける。ここで、終了タグ処理部３４の処理内容については、後述する。

【００３３】また処理Ｓ１３において、終了タグでなければ（Ｎｏならば）、通常のコンテンツとして処理を行い（Ｓ１５）、処理Ｓ３に戻って処理を続ける。

【００３４】ここで、ＣＰＵ１１が行う開始タグ処理部３３の動作について説明する。ＣＰＵ１１は、開始タグ処理部３３を読み込まれた文書データを引数として起動し、まず、読み込まれたタグの要素を木構造の現時点でポイントしている要素の１レベル下位に追加し、追加した要素を新たに現在の要素としてポイントする。また最初のタグであれば（木構造がなければ）、この最初のタグを木構造の最初のレベル（ルート）として登録し、当該ルートを現在の要素としてポイントする。

【００３５】具体的に図４に示した文書データに対し、ＣＰＵ１１は、図７に示す木構造を形成する。すなわち、図４の文書データの１行目の<HTML>を開始タグとして認識し、開始タグ処理部３３の処理としてこの「HTML」を木構造のルートとして設定してポイントし（Ｘ）、次に２行目の<HEAD>をさらに開始タグとして認識して現在ポイントしている要素（HTML）より１レベル下位に「HEAD」の要素を追加して（Ｙ）、この「HEAD」をポイントする。さらに３行目の「TITLE」も開始タグであるので、さらに１レベル下位に「TITLE」の要素を付加し、引き続く「HOME PAGE」をＰＣＤＡＴＡとして付加する（Ｚ）。

【００３６】また、ここで、ＣＰＵ１１が行う終了タグ処理部３４の動作について説明する。ＣＰＵ１１は、終了タグ処理部３４を読み込まれた文書データと木構造中で現在ポイントしている要素とを引数として起動し、ＸＭＬの木構造の要素をＨＴＭＬとして解析する処理を行って、当該要素をＨＴＭＬ文書に変換してＲＡＭ１３に格納し、木構造中でポイントしている位置を１レベル上位に設定して、新たに現在の要素としてポイントする。具体的に図４及び図７においては、「TITLE」要素に付加されているＰＣＤＡＴＡ「HOME PAGE」（Ｚ）の後の「</TITLE>」を終了タグとして認識し、開始タグ「<TITLE>」から終了タグ「</TITLE>」までの基本構造「<TITLE>HOME PAGE</TITLE>」をＨＴＭＬ文書として解析し、ＨＴＭＬ文書としてＲＡＭ１３に格納する。そして、「TITLE」より１レベル上位の「HEAD」（Ｙ）をポイントするようにして、終了タグの処理を完了する。このようにして、終了タグ処理部３４の処理の結果として得られるＨＴＭＬ文書は、図８に示すように、行構造に変換されたものとなる。

【００３７】さらにＣＰＵ１１は、ＸＭＬパーズ部２２の処理が完了すると、ブラウザコア部２４を起動して、ＲＡＭ１３に格納されたＨＴＭＬ文書（ＨＴＭＬ及びＸＭＬの混在した文書からＸＭＬパーズ部２２及びＨＴＭＬパーザ部２３の動作によりＨＴＭＬに変換された文書）を参照し、このＨＴＭＬ文書に基づいて描画部２５に描画の指示を出力する。描画部２５は、文書の各行の絶対座標やサイズ、テキストや画像部分、絶対座標やサイズ、テキストに対するフォント、画像を参照する情報としてのＵＲＬ（Uniform Resource Locators）等を考慮して、ディスプレイ１５に描画結果としての文書データの内容を表示する。

【００３８】尚、ここまでの説明においては、デフォルトＤＴＤを予めロードしているが、ＸＭＬタグ処理において、データ型を宣言するタグである「<!DOCTYPE」が検出されたときに、当該データ型が特定のもの（例えばＨＴＭＬ３．２）であったときにのみデフォルトＤＴＤをロードするようにしてもよい。また、開始タグ処理部３３の動作として、「<HTML>」タグを検出した場合に、それまでにＤＴＤ宣言（「<!DOCTYPE」）を検出していないときにのみ、デフォルトＤＴＤをロードすることとしてもよい。

【００３９】さらに、読み込まれた文書データに関連して、当該文書データ内部又は当該文書データから参照されるＤＴＤがある場合には、ＣＰＵ１１は、このＤＴＤをロードして、デフォルトＤＴＤの代わりに、又はデフォルトＤＴＤとともにＸＭＬパーズ部２２の処理において利用することも好適である。

【００４０】また、動作パラメータにおいて、終了タグの省略の可否を表すフラグにより、特定のタグの省略が可能と設定された場合には、ＣＰＵ１１は、当該開始タグを読み込むと、木構造に追加された要素の１つ上位の要素を取得しておき、ロードされたＤＴＤを参照して、省略可能であるときには、タイプがＥＭＰＴＹである（「/>」で終了するタグである）か、又はＥＭＰＴＹでないが要素宣言中コンテンツに含まれる可能性のあるタグのリストにないタグであるときに、終了タグが省略されたものとして終了タグ処理部３４を起動する。

【００４１】また、省略可能でないタグであるときや、ＥＭＰＴＹでなく、かつ要素宣言中コンテンツに含まれる可能性のあるタグのリストにある場合には、当該タグを木構造に追加して開始タグ処理部３３を起動する。

【００４２】

11

【発明の効果】本発明によれば、XMLパーズ手段が、必要に応じてHTMLパーズ手段を起動するので、文書データを効率的に処理できる。

【００４３】また本発明によれば、デフォルトデータ型定義が利用されるので、データ型定義に関連づけられていない文書データも正常に処理できる。

【００４４】さらに、本発明によれば、省略可能なタグがある場合に、終了タグを補完して処理するので、タグを省略しても正常に処理できる。

【図面の簡単な説明】

【図１】　本発明の実施の形態に係る文書データ処理装置の構成ブロック図である。

【図２】　ＣＰＵ１１が処理するブラウザのソフトウェアの構造を表す構成ブロック図である。

【図３】　文書データの一例を表す説明図である。

12

＊【図４】　ＣＰＵ１１のXMLパーザとしての処理を表すフローチャート図である。

【図５】　動作パラメータの一例を表す説明図である。

【図６】　ＤＴＤの一例を表す説明図である。

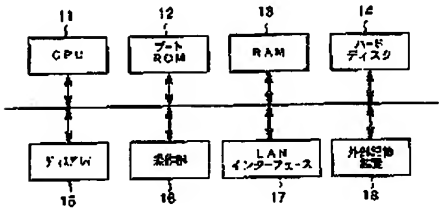
【図７】　木構造の一例を表す説明図である。

【図８】　行構造の一例を表す説明図である。

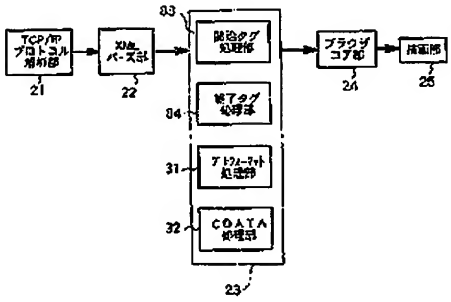
【符号の説明】

１１　ＣＰＵ、１２　ブートROM、１３　ＲＡＭ、１４　ハードディスク、１５　ディスプレイ、１６　操作部、１７　ＬＡＮインタフェース、１８　外部記憶装置、２１　ＴＣＰ／ＩＰプロトコル解析部、２２　ＸＭＬパーズ部、２３ＨＴＭＬパーズ部、２４　ブラウザコア部、２５　描画部、３１　プレフォーマット処理部、３２　ＣＤＡＴＡ処理部、３３　開始タグ処理部、３４　終了タグ処理部。

【図１】



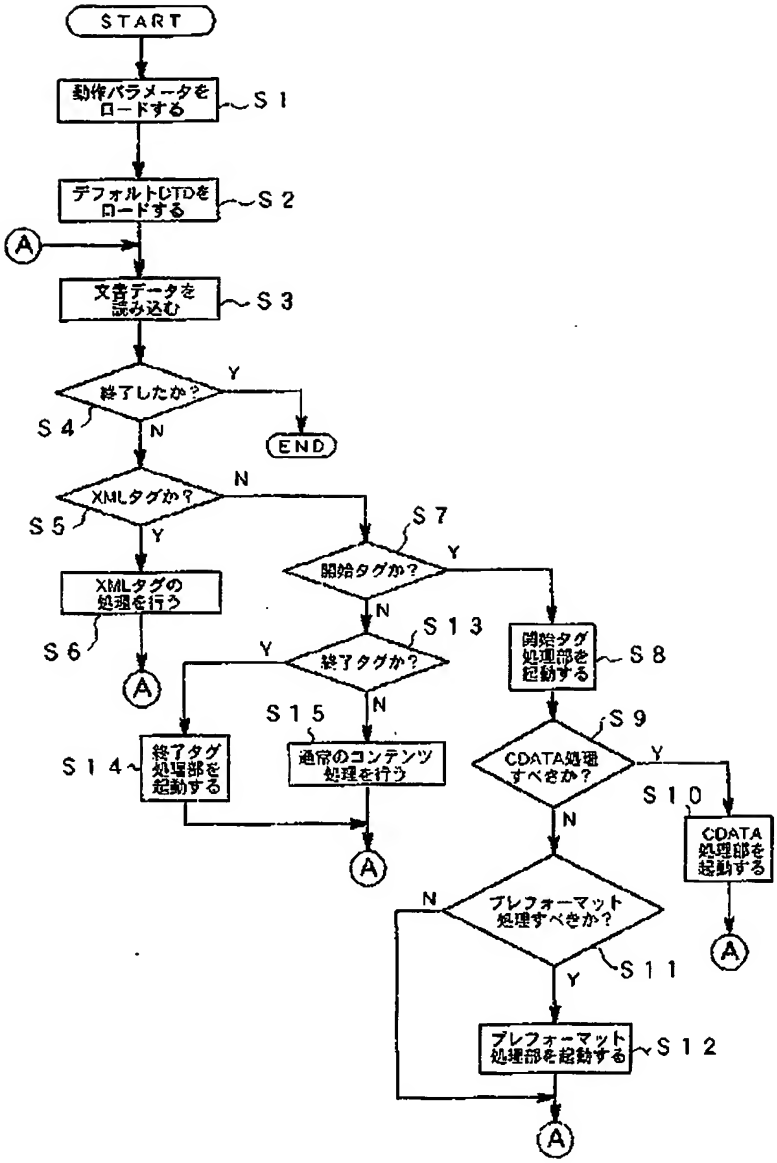
【図２】



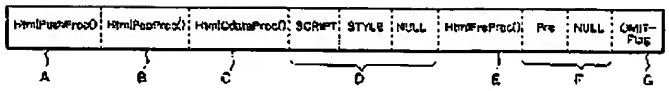
【図３】

```
<HTML>
<HEAD>
<TITLE> HOME PAGE </TITLE>
<STYLE>
H1  { font : bold 20pt serif}
P   { font : 12pt times}
</style>
</HEAD>
<BODY>
<CENTER>
<H1> Welcome to ××××××××× </H1>
<BR>
<P>
<A href=" http://××× ×××.××/jp/jp/ ><IMG SRC=" enter.gif" >  </A>
<BR>
<A href=" http://×××.×××.××/jp/jp/" > Click here to enter. </A>
<BR>
</CENTER>
<PRE>

        +-----------------------------+
        | (C) ×××× Text Data          |
        +-----------------------------+


</PRE>
</BODY>
</HTML>
```

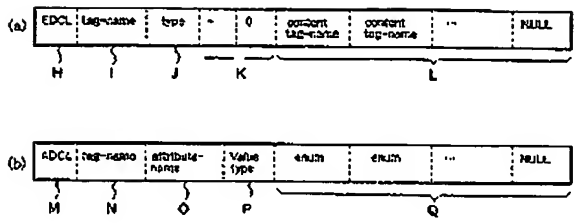
【図4】


START
動作パラメータをロードする
S1
デフォルトDTDをロードする
S2
A
文書データを読み込む
S3
終了したか？
S4
Y
N
END
XMLタグか？
S5
N
Y
XMLタグの処理を行う
S6
A
S7
開始タグか？
Y
N
S13
終了タグか？
Y
N
S14
終了タグ処理部を起動する
S15
通常のコンテンツ処理を行う
A
開始タグ処理部を起動する
S8
S9
CDATA処理すべきか？
Y
N
S10
CDATA処理部を起動する
A
プレフォーマット処理すべきか？
N
Y
S11
プレフォーマット処理部を起動する
S12
A

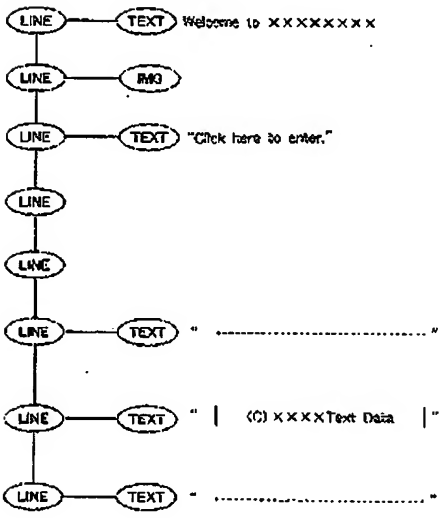
【図5】


HtmlPreProc()
HtmlPostProc()
HtmlCdataProc()
SCRIPT
STYLE
NULL
HtmlPreProc()
Pre
NULL
OMIT-Flag
A
B
C
D
E
F
G

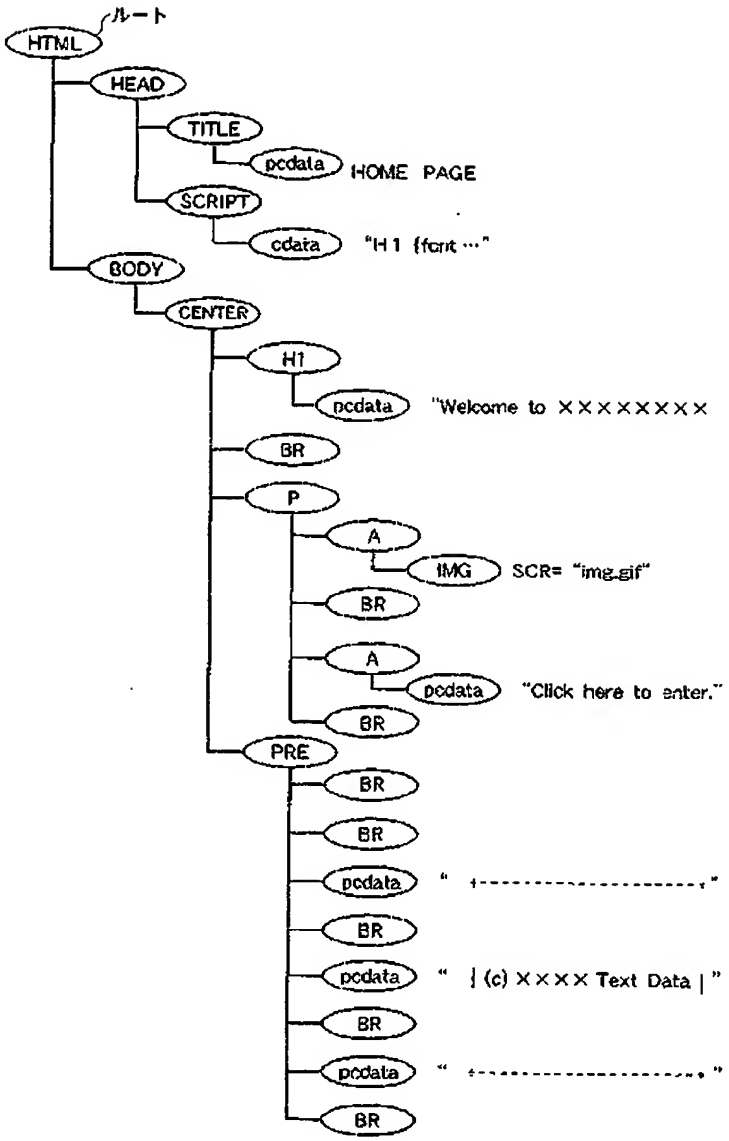
【図６】


(a)
EDCL
tag-name
type
~
0
content tag-name
content tag-name
...
NULL
H
I
J
K
L
(b)
EDCL
tag-name
attribute-name
Value type
enum
enum
...
NULL
M
N
O
P
Q

【図８】


LINE
TEXT
Welcome to ××××××××
LINE
IMG
LINE
TEXT
"Click here to enter."
LINE
LINE
LINE
TEXT
" ----------"
LINE
TEXT
" | (C) ××××Text Data |"
LINE
TEXT
" ----------"

【図7】


ルート
HTML
HEAD
TITLE
pcdata
HOME PAGE
SCRIPT
cdata
"H1 {font …"
BODY
CENTER
H1
pcdata
"Welcome to ××××××××
BR
P
A
IMG
SCR= "img.gif"
BR
A
pcdata
"Click here to enter."
BR
PRE
BR
BR
pcdata
" +----------+ "
BR
pcdata
" | (c) ×××× Text Data | "
BR
pcdata
" +----------+ "
BR